Artificial Intelligence and Machine Learning Journal, ISSN: 1687-4846, Vo. 17, No. 1, Delaware, USA, October 2017

www.icgst.com

Face Tracking using Microsoft Xbox One Kinect and Mimicking Robotic Neck

Jimdelle C. Kintanar, Russel Roman L. Mallari, Jasmin D. Maullon, Ma. Camille H. Ragual, Julius M. Redilla and Roselito E. Tolentino
Polytechnic University of the Philippines-Santa Rosa Campus, Philippines
jimdelle.kin027@gmail.com, RSL_mallari@yahoo.com, maullon.jasmin@gmail.com, camille_ragual@yahoo.com, redilla16julius@gmail.com, and kenmetara@yahoo.com

Abstract

This study aims to control a robotic neck with five degree of freedom using the face tracking feature of Microsoft Kinect Xbox One for different users. Past studies involved the use of wearable devices as a controller limited to only one person. On those studies, it is observed that the capability of the user to move on a certain angle is affected by using such wearables. To resolve the problem, the authors used the Microsoft Kinect Xbox One as a controller. This study implements the Quaternion concept used in the face tracking algorithm of Microsoft Kinect. Upon collecting the parameters, conversion to Euler angles are executed for ease of controlling the prototype. The acquired data are then evaluated using the z-test method. The authors used the functionality of the LabVIEW to graph and extract the data used in statistics. The authors conclude that 1) the proposed system is effective in acquiring the necessary neck angle for controlling the robotic neck and 2) the actual neck angle is close to the robotic neck angle.

Keywords: *Face tracking, Microsoft Kinect, Mimicking, Robotic neck, Five degree of freedom, Different users.*

Nomenclature

LabVIEW	Laboratory Virtual Instrument Engineering Workbench
LIFA	LabVIEW Interface for Arduino
SDK	Software Development Kit
H_0	Null Hypothesis
H_1	Alternative Hypothesis
α	Significance level

1. Introduction

In recent years, imitation and other forms of social learning hold tremendous promise as a powerful means for robots to acquire new tasks and skills. The need for creating such robots that mimic arises from the diverse socio-economic interests in nearly inaccessible areas such as mines, and areas exposed to radiation, war zones, etc. Some researchers aimed to mimic the human body or a part of it, but most of studies engrossed on imitating or mimicking human body parts like the arm, leg, or shoulder. [1][2] However, few researchers also engaged in studying robots that can mimic the human neck. Experts on this field are constantly conducting studies about the ability of the neck to widen the application of the robotic neck. Most of the studies conducted used robotic neck for medical purposes. [3] Some used a wearable controller embedded with a potentiometer, accelerometer, gyroscope, or any sensor that sends data to the actuator, to control the robot. To prolong the life of the system with this kind, various groups have attempted to accurately translate human motions in a robot using image processing. [4-7]

A recent study presented in [8] used wearable controller to mimic the human neck. To obtain values for the neck angle the researchers used a wearable controller embedded with potentiometer. Researchers integrated these values to the LabVIEW to communicate with the microcontroller. But it has been identified that the main limitation of this project is the controller. In this system, the controller is limited only to one user. It is also observed that the capability of the user to move on a certain angle is affected by the controller.

To overcome the limitation of the previous study, the authors used the Microsoft Kinect Xbox One and created an algorithm that can track the face and can be used by different users. The face tracking feature of Kinect SDK is used to obtain the angles made by the human neck in five degree of freedom. For the face tracking, Quaternion approach is used in the algorithm and the parameterization of Quaternion like w, x, y, and z is used to extract the neck angles for extension/flexion, lateral bending, rotation, hyperextension/flexion and hyper lateral bending.

In this study, the detection of the Microsoft Kinect Xbox One focuses on the head. The robotic neck can perform

10° for flexion, 30° for extension, 30° for lateral bending on both sides (left and right), and 45° for the rotation (left and right). For the hyper movements, the robot can perform 30° for flexion, 10° for extension, and 15° for lateral bending on both sides (left and right).

2. Methodology

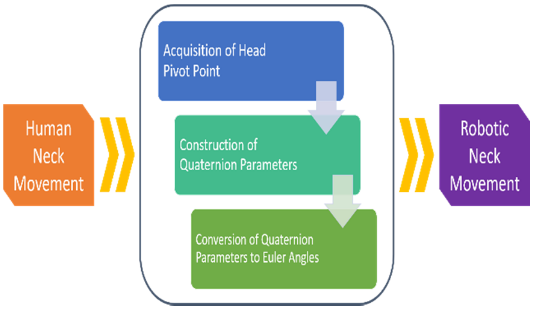


Figure 1. Conceptual Framework

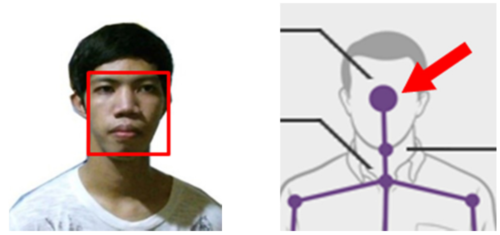
Figure 1 shows the general idea on how the system works. The features of Microsoft Kinect Xbox One are used to acquire the angles needed to mimic human neck.

The authors used the concept of face tracking by using C++ language in Microsoft Visual Studio. To show that a face is being tracked, a bounding box that contains the head pivot point is displayed in the screen. The parameterization of Quaternion like w, x, y, and z is used to extract the neck angles. Using the function of this feature, the angles of the neck extension/flexion, lateral bending, rotation, hyperflexion/extension, and hyper lateral bending are achieved. The authors are able to get the angles by applying the Quaternion concept.

The acquired angles are processed in Microsoft Visual Studio and communicated with the microcontroller. The microcontroller then inputs the required angle to the servo motors of the robotic neck, enabling to mimic the movement of the human neck. The authors used LabVIEW to view the response of the system.

A. Neck Angle Acquisition

Acquisition of head pivot point: First, to be able to obtain the needed angles produced by the algorithm, the authors have used the face tracking feature of Microsoft Kinect for Xbox One which retrieved the points in infrared and colour space. Through the use of Microsoft Visual Studio and Kinect SDK, the authors are able to create a program in C++ language to acquire the necessary values for neck movement angles.

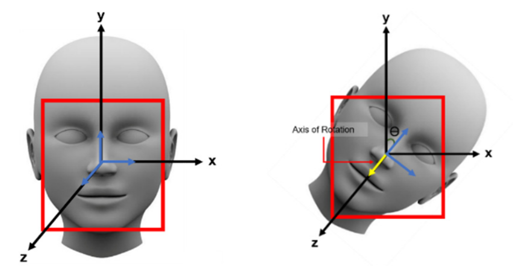


a b

Figure 2. (a) Face being track (b) head pivot point

For the analysis of the neck movement angles, Microsoft Kinect will tracks the face of the user. Face detection displays a bounding box (Figure 2a), which is a rectangle that defines the user's head, as determined by the face detection algorithms. The head pivot point (Figure 2b), obtained through FaceRotationQuaternion property, is the computed center of the head, within which the face may be rotated around. This point is defined in the Kinect body coordinate system. The head pivot point is comparable to the head joint of the body, but the head pivot point has a different vertical coordinate (suitable as a center of rotation).

Construction of Quaternion Parameters: Quaternion is a four-dimensional complex number composed of one real dimension and three imaginary dimensions.



a b

Figure 3. (a) Face is being tracked (b) Face is being rotated

Figure 3a shows the head track by the bounding box and the axis involved with it. The bounding box is stationary in the face of the user and serves as reference frame. The axis of the tracked face is then rotated as shown in Figure 3b. It shows that as the axis of the face rotates in relation with the reference frame, an angle of rotation is made. The axis of rotation is the yellow section perpendicular to the movement of the head. From these values, coordinates will be acquired.

$$\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} \cos(0.5\theta) \\ \upsilon_x \sin(0.5\theta) \\ \upsilon_y \sin(0.5\theta) \\ \upsilon_z \sin(0.5\theta) \end{bmatrix}$$

Where
a.b.c.d are Quaternion elements
$\upsilon_x, \upsilon_y, \upsilon_z$ *are vectors*
θ *is the angle of rotation*

Figure 4. Conversion of vector to quaternion elements

Figure 4 shows the conversion of axis of rotation (vector) to quaternion. This conversion is done in the Visual Studio code. The coordinates for axis of rotation (vector) are computed by Kinect through its coordinate system. The acquired vectors are converted to quaternion elements w, x, y, and z. These quaternion parameters are then converted to Euler angles.

Conversion of Quaternion to Euler Angles: The quaternions acquired give an appropriate data to determine the movement of the head. These quaternions are then converted into Euler angles. Euler angles provide a way to represent the 3D orientation of an object using a combination of three rotations about different axes. With this, the angles for pitch, yaw, and roll of the neck can be obtained. Conversion of quaternion is done by writing the formula for Euler angle in the code.

For Flexion/Extension:

$$Pitch = \left(\arctan\left(\frac{2(yz + wx)}{w^2 - x^2 - y^2 + z^2} \right) \right) \times \frac{180}{\pi} \quad (1)$$

For Rotation:

$$Yaw = \left(\arcsin\left(2(wy - xz) \right) \right) \times \frac{180}{\pi} \quad (2)$$

For Lateral Bending:

$$Roll = \left(\arctan\left(\frac{2(xy + wz)}{w^2 + x^2 - y^2 - z^2} \right) \right) \times \frac{180}{\pi} \quad (3)$$

For Hyperflexion/extension:

$HPitch = Pitch + 10$; if Pitch is negative

$HPitch = Pitch - 10$; if Pitch is positive

For Hyper Lateral Bending:

$HRoll = Roll + 30$; if Roll is negative

$HRoll = Roll - 30$; if Roll is positive

Where w, x, y and z are the quaternion elements, and Pitch, Yaw, Roll, HPitch and HRoll corresponds with the computed neck angles.

The computation of hyper values is the result of subtracting the computed normal movements versus the fixed angles designated by the authors. The angles designated are close to the actual angles made by the human neck. This is done to accomplish the five-degree-freedom robotic neck and since Microsoft Kinect only outputs yaw, pitch, and roll.

B. Evaluating the Significant Difference between the Human Neck Movements and the Robotic Neck Movements

A wearable helmet is devised in order to assess the response of the system. The wearable helmet is composed of five potentiometers used to acquire the actual neck angle for five degree of freedom. The angular displacement on the potentiometers created by the neck's movement is sent to the microcontroller for interpretation. The microcontroller, which is the Arduino, is used to send the data serially to the computer which is processed by the LabVIEW. This software is used by the authors to simulate and evaluate the data gathered. The communication between the LabVIEW and the Arduino is made possible through LabVIEW Interface for Arduino (LIFA).
In order to actuate the robotic neck (Figure 5), seven servo motors (for flexion and extension, lateral bending, lateral rotation, hyperflexion and hyperextension, and hyper lateral bending) are used. These actuators are mounted on the neck joint, and a Kinect Xbox One is used as a sensor. The servo motors are connected on the Arduino and are controlled through Microsoft Visual Studio.

Figure 5. Demonstration of the robotic neck

After getting the angular data from the user wearing the helmet and the robotic neck, data are plotted with LabVIEW. The gathered and computed data are input into Excel Worksheet where the evaluation is made. A z-test is applied by the authors to evaluate the response of the control system using the acquired angles. In using z-test, it is necessary to define the null hypothesis (H_0), alternative hypothesis (H_1), and the critical value to prove that the hypothesis is true.

To know the critical value for a two-tailed test, the significance level (α) is set to 5%. Setting this significance value creates a confidence of 95% (obtained by 100% − α), and sets 0.975 as the area of the curve as the critical value (obtained by 1 – (α/2)). Knowing the area, the authors used the z-test table (area under the normal curve) and found the critical value 1.96.

	Hypothesis	Condition
Null (H_0)	There is no significant difference between the human neck movement and the robotic neck movement.	$-1.96 \leq z \leq 1.96$
Alternative (H_1)	There is a significant difference between the human neck movement and the robotic neck movement.	$-1.96 > z > 1.96$

Table 1. Statement of the Hypothesis

Table 1 shows the null and alternative hypotheses and the conditions to be accepted for z-test. The computed z value is the basis of the authors to accept or reject the hypothesis. If the z value is between the ranges of -1.96 to +1.96, the null hypothesis is accepted; otherwise, the alternative hypothesis is accepted. To obtain the z value, the authors used the following formula:

$$z = \frac{\overline{x_1} - \overline{x_2}}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (4)$$

Where:
z = z-test result
n_1 = number of samples in the first sample group
n_2 = number of samples in the second sample group
$\overline{x_1}$ = mean value of the first sample group
$\overline{x_2}$ = mean value of the second sample group
σ_1 = standard deviation of the first sample group
σ_2 = standard deviation of the second sample group

3. Results and Discussion

A. Neck Angle Acquisition

For the data gathering, the authors accomplished different neck movements. The users performed flexion/extension, lateral bending, rotation, hyperflexion/extension and hyper lateral bending.

To test if the controller and the human neck have the same or close value, five users performed flexion/extension with five-degree intervals and got the angle difference between them. The authors captured the movements and measurements through the user interface of the program, then recorded the values obtained from the actual and the computed angles. Summary of results are shown in Tables 2–6.

User	Angle Difference
User 1	0.1289
User 2	0.1498
User 3	0.1839
User 4	0.1250
User 5	0.1229
Average	**0.1421**

Table 2. Summary of the average angle differences of the five users in terms of flexion/extension

User	Angle Difference
User 1	0.0975
User 2	0.1520
User 3	0.1399
User 4	0.1835
User 5	0.1664
Average	**0.1479**

Table 3. Summary of the average angle differences of the five users in terms of lateral bending

User	Angle Difference
User 1	0.1275
User 2	0.1968
User 3	0.1485
User 4	0.1711
User 5	0.1641
Average	**0.1616**

Table 4. Summary of the average angle differences of the five users in terms of rotation

User	Angle Difference
User 1	0.1454
User 2	0.1641
User 3	0.1382
User 4	0.1383
User 5	0.2040
Average	**0.1580**

Table 5. Summary of the average angle differences of the five users in terms of hyperflexion/extension

User	Angle Difference
User 1	0.1337
User 2	0.1874
User 3	0.1971
User 4	0.1862
User 5	0.1590
Average	**0.1727**

Table 6. Summary of the average angle differences of the five users in terms of hyper lateral bending

Tables 2–6 show the summary of the angle difference of the five users in terms of hyper lateral bending. It is indicated that the neck movements have the average angle difference of 0.1421°, 0.1479°, 0.1616°, 0.1580° and 0.1727°, respectively. All the values are close to zero. It means that there is little difference between the actual and the computed neck angles. This is due to the unsteady detection of the Kinect sensor while capturing

the movement. The sensor constantly tracks the face in real time and a slight movement causes some minor inconsistency. Nevertheless, the values of each computed neck angle are close to the neck angles of the user.

B. Evaluation of the Significant Difference between the Human Neck Movements and the Robotic Neck Movements

In the experiment, the users have performed random movements to test the capability and the accuracy of the prototype to mimic flexion/extension, lateral bending, rotation, hyperflexion/extension, and hyper lateral bending. To obtain data, the authors have created a wearable device that is adjustable for the user and placed potentiometers in prototype joints. The results are plotted in a graph using LabVIEW and tabulated using Microsoft Excel. The sample graph is shown in Figure 6.

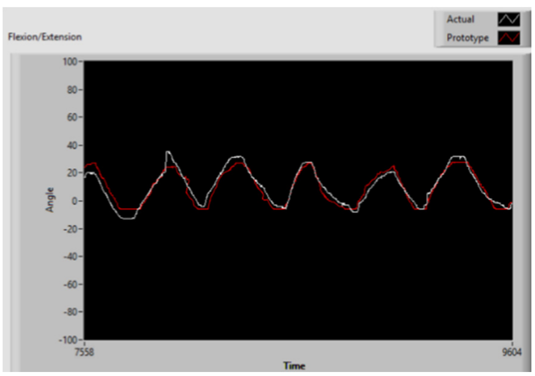


Figure 6. LabVIEW panel showing the system response of flexion/extension

Figure 6 show the sample response of the actual and the prototype angles for flexion/extension. This graph shows the angle being measured from the wearable device represented by white lines and the robotic neck represented by red lines. It is observed that the white and red lines in the graphs are close to each other; and that there are more jitters in red lines than in white lines. This indicates that the robotic neck creates more noise caused by the prototype's mechanism.

The researchers evaluated the angular data taken from five different movements performed by the user and the robotic neck. The data used are acquired from the graph of LabVIEW.

Table 7 shows the z-test evaluation of the angular data gathered from the actual human neck and robotic neck for different users. It is composed of number of samples (n1 and n2), the mean of the samples ($\bar{x}1$ and $\bar{x}2$), the standard deviation of the samples (σ1 and σ2) and the z-test result. As presented in the table, the authors have observed that the robotic neck is successful in mimicking movements of the human neck. They have also noted that the hyper movements yielded some of the highest z-values. This is caused by the unsteady detection while the user is performing the movement and the occasional loss of detection when in critical angle. Moreover, the prototype's mechanism where the hyper movements components hold the weight of the upper performing components influenced the result.

Overall, all the z-values are within the acceptance region. Table 7 shows that the z-test results are within the range of -1.96 to +1.96. Thus, the null hypothesis that "there is no significant difference between the human neck movement and the robotic neck movement." is accepted. Meaning, the robotic neck angles are close to the actual user angles.

4. Conclusion

Based on the data gathered, the authors calculated the angular difference between the actual neck angle and angle produced by the algorithm to test the accuracy. The values are close to each other with the difference ranging from 0.1421° to 0.1727°. Lateral bending had the lowest average angular difference with the value 0.1421° making it the most reliable reading of the Kinect sensor. On the other hand, hyper lateral bending had the highest average angular difference with the value of 0.1727°, making it the least reliable. Although it is similar to the normal lateral bending there are times that the detection is loss when performing the critical angle. Overall, the result from the angular difference suggests that the angles do not differ significantly in five different users; thus, the algorithm proposed by the researchers was effective in acquiring the value of the user's neck angle.

In terms of the statistical data gathered by the authors, all of the z-test results for flexion/extension, lateral bending, rotation, hyper flexion/extension, and hyper lateral bending were within the range of -1.96 and +1.96. Therefore, the authors concluded that the prototype can effectively mimic the user's movements. Although the z-test values fall within the acceptance region, it can be observed that the other movements had large but tolerable angular difference. This was because the sensor is constantly tracking the face in real time and a little movement causes small inconsistency. For the hyper movements, it holds the weight of the upper components.

5. Recommendation

The authors recommend to improve the proposed algorithm or introduce new algorithm that can track the face effectively even when performing hyper movements. The authors also recommend to improve the working prototype that can handle the actual weight of the head to perform the five degree of freedom efficiently.

	Movements	Actual Angle			Prototype Angle			Results of z-test	Remarks
		n_1	$\bar{x}_1$	σ_1	n_2	$\bar{x}_2$	σ_2		
USER 1	Flexion / Extension	100	11.11	13.49	100	9.77	13.43	0.70	Null Hypothesis is Acceptable
	Lateral Bending	100	18.19	9.37	100	17.49	8.83	0.54	
	Rotation	100	19.66	12.32	100	20.64	14.36	-0.52	
	Hyper Flexion / Extension	100	-14.73	7.51	100	- 13.54	8.40	-1.06	
	Hyper Lateral Bending	100	7.28	2.57	100	6.92	2.94	0.92	
USER 2	Flexion / Extension	100	3.15	8.27	100	2.32	8.25	0.71	Null Hypothesis is Acceptable
	Lateral Bending	100	17.85	8.91	100	18.87	8.56	-0.83	
	Rotation	100	-20.16	10.62	100	- 21.67	10.87	0.99	
	Hyper Flexion / Extension	100	-14.99	6.63	100	- 13.66	6.87	-1.39	
	Hyper Lateral Bending	100	7.27	3.58	100	7.99	5.21	-1.14	
USER 3	Flexion / Extension	100	7.39	10.19	100	6.41	12.82	0.60	Null Hypothesis is Acceptable
	Lateral Bending	100	18.70	9.54	100	17.67	9.12	0.78	
	Rotation	100	25.84	7.55	100	27.19	9.57	-1.11	
	Hyper Flexion / Extension	100	-18.32	6.33	100	- 17.20	6.91	-1.20	
	Hyper Lateral Bending	100	-3.84	2.53	100	-4.15	3.29	0.75	
USER 4	Flexion / Extension	100	10.69	6.44	100	11.56	6.49	-0.95	Null Hypothesis is Acceptable
	Lateral Bending	100	10.28	12.32	100	11.40	12.68	-0.63	
	Rotation	100	-21.35	12.48	100	- 23.61	12.47	1.28	
	Hyper Flexion / Extension	100	-12.72	6.46	100	- 13.39	10.19	0.56	
	Hyper Lateral Bending	100	-6.39	3.72	100	-5.72	3.02	-1.40	
USER 5	Flexion / Extension	100	-11.06	7.73	100	- 10.37	7.99	-0.62	Null Hypothesis is Acceptable
	Lateral Bending	100	14.21	9.34	100	13.36	7.89	0.70	
	Rotation	100	-19.15	9.02	100	- 20.70	10.71	1.11	
	Hyper Flexion / Extension	100	-10.63	4.47	100	- 11.48	6.49	1.08	
	Hyper Lateral Bending	100	-6.48	2.11	100	-6.14	2.73	-0.99	

Table 7. Z-test result for varying angle in each degree of freedom for different users

6. References

[1] Rudio, DJ. C., et al. (2016) Application of Microsoft Xbox One for Mimicking Robotic Wrist with Three Degree of Freedom in Different Poses. International Journal of Engineering Research Volume No.5, Issue No.4, pp: 253–259

[2] Ortega RJ. R., et al. (2016) Vector Multiplication Approach for Point of View Variations in a Mimicking Robotic Shoulder Using Microsoft Kinect. International Journal of Inventive Engineering and Sciences (IJIES) ISSN: 2319–9598, Volume-4 Issue-1

[3] Amanda H.A.H., et al. (2015). Interactive Robotic Head for Neck Exercises. 2015 Moratuwa Engineering Research Conference (MERCon)

[4] Tölgyessy, M., & Hubinský, P. (2011). The Kinect Sensor in Robotics Education. 2nd International Conference on Robotics in Education (RiE 2011). Vienna, Austria, September, 2011. pp. 143-146.

[5] Brouwer, D.M., Bennik, J., Leideman, J., Soemers, H.M.J.R., & Stramigioli, S. (2009). Mechatronic Design of a Fast and Long Range Four Degrees of Freedom Humanoid Neck. 2009 IEEE International Conference on Robotics and Automation

[6] Hackel, M., Schwope, S., Fritsch, J., Wrede, B., & Sagerer, G. (2005). Humanoid robot platform suitable for studying embodied interaction. 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems

[7] Nori, F., Jamone, L., Sandini, G., & Metta, G. (2007). Accurate Control of a Human-like Tendon-driven Neck. 2007 7th IEEE-RAS International Conference on Humanoid Robots

[8] Benitez, E. D., et al. (2016). Design and Implementation of Mimicking Robotic Neck Improving Hyper-flexion/extension Movement and Lateral Bending using Linear Motor for Force Requirement Improvement. International Journal of Innovative Technology and Exploring Engineering

Biographies

Jimdelle C. Kintanar is presently living in Sta. Rosa City, Laguna. He is currently taking bachelor's degree in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa, Laguna Campus. He was part of NXP Semiconductors Cabuyao Inc., Philippines as a team member of Sensors-Plating Department for his on-the-job training in 2015. He was part SkyCable Corporation, Philippines as a team member of Engineering Department for his on-the-Job training in 2016. He is currently a member of Association of Electronics and Communications Engineering and Institute of Electronics Engineers of the Philippines.

Russel Roman L. Mallari is presently living in Sta. Rosa City, Laguna. He is currently taking bachelor's degree in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa, Laguna Campus. He was part of NXP Semiconductors Cabuyao Inc., as a team member of Sensors-Plating Department for his on-the-Job training in 2015. He was part Philippine Long Distance Telephone─LTI, as a team member of Main Distribution Frame Department for his on-the-Job training in 2016. He is currently a member of AECES and IECEP.

Jasmin D. Maullon is presently living in Carmona, Cavite. She is currently taking bachelor's degree in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa, Laguna Campus. She was part of Efficient Maschinentechniks Inc., as a team member of Engineering Department for her on-the-Job training in 2015. She was part Total Powerbox Solutions Inc., as a team member of Engineering Department for her on-the-Job training in 2016. She is currently a member of AECES and IECEP.

Ma. Camille H. Ragual is presently living in Biñan, Laguna. She currently taking bachelor's degree in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa, Laguna Campus. She was part of Sensor Scientific Philippines, Inc., Philippines as a team member of Engineering Department for her on-the-Job training in 2015. She was part Total Powerbox Solutions, Inc., as a team member of Engineering Department for her on-the-Job training in 2016. She is currently a member of AECES and IECEP.

Julius M. Redilla is presently living in Biñan, Laguna. He is currently taking bachelor's degree in Electronics and Communications Engineering from the Polytechnic University of the Philippines Santa Rosa, Laguna Campus. He was part of TDK Philippines Corporation, as a team member of Engineering Department for his on-the-Job training in 2015. He was part Pilipino Cable Corporation, as a team member of Engineering Department for his on-the-Job training in 2016. He is currently a member of AECES and IECEP.

Roselito E. Tolentino is a registered Electronics Engineer and IECEP-Member. He is a graduate of B.S. Electronics and Communications Engineering at Adamson University in 2004 with a scholarship from DOST-SEI. He finished his Master of Science in Electronics Engineering Major in Control System at Mapua Institute of Technology with a scholarship from DOST-ERDT. He currently takes up Doctor of Philosophy in Electronics Engineering at the same institute. He is currently working as a part-time instructor at the Polytechnic University of the Philippines Santa Rosa Campus and De La Salle University Dasmariñas. His research interests are more on Robotics and Machine Vision.